

Capítulo VINTE

Asserções

Objetivos do Exame

Testar invariantes usando asserções.

Usando asserções

Uma asserção é uma instrução usada para verificar se algo é verdadeiro e ajuda você a detectar erros em um programa.

Uma instrução assert tem a seguinte sintaxe:

```
java Copiar Editar

assert expressãoBooleana;
```

Se a expressãoBooleana for avaliada como FALSA, uma exceção do tipo `java.lang.AssertionError` (uma subclasse de `Error`) é lançada.

Em outras palavras, a instrução acima é equivalente a:

```
java Copiar Editar

if (expressaoBooleanaFalsa) {
    throw new AssertionError();
}
```

Isso acontece porque, como dito antes, uma asserção precisa ser verdadeira. Se não for, um erro deve ser lançado.

Uma instrução assert também pode receber uma String como mensagem:

```
java Copiar Editar

assert expressãoBooleana: "Mensagem sobre o erro";
```

Que é equivalente a:

```
java Copiar Editar

if (expressaoBooleanaFalsa) {
    throw new AssertionError("Mensagem sobre o erro");
}
```

A questão é que as asserções NÃO estão habilitadas por padrão.

Você pode ter asserções em todo o seu código, mas se elas estiverem desabilitadas, o Java as ignorará completamente.

Em resumo:

- Se as asserções estiverem habilitadas e a expressão booleana for verdadeira, nada acontece.
- Se as asserções estiverem habilitadas e a expressão booleana for falsa, um `AssertionError` é lançado.
- Se as asserções estiverem desabilitadas, não importa o resultado da expressão booleana, elas são ignoradas.

As asserções são habilitadas na linha de comando com:

```
bash Copiar Editar
java -ea ClassePrincipal
```

Ou:

```
bash Copiar Editar
java -enableassertions ClassePrincipal
```

Isso habilitaria as asserções em todas as classes do nosso programa, exceto nas classes do Java (ou classes do sistema).

Por exemplo, se tivermos esta classe:

```
java Copiar Editar

public class TestAssertion {
    public static void main(String[] args) {
        // Parênteses são opcionais
        assert (args.length > 0) : "Pelo menos um argumento é necessário";
        System.out.println(args[0]);
    }
}
```

E executarmos assim:

```
bash Copiar Editar
java -ea TestAssertion
```

A saída será:

```
php Copiar Editar

Exception in thread "main" java.lang.AssertionError: Pelo menos um argumento é necessário
```

Se executarmos assim:

```
bash Copiar Editar
java -ea TestAssertion Oi
```

A saída será:

```
nginx Copiar Editar

Oi
```

Se executarmos assim:

```
bash Copiar Editar
java TestAssertion
```

A saída será:

```
cpp Copiar Editar

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
```

Como você pode ver, as asserções podem nos ajudar a capturar alguns erros derivados de suposições que fazemos em nosso programa. Apenas esteja ciente de que elas não estão habilitadas por padrão. Elas não são projetadas para serem executadas em produção; para isso, é melhor usar exceções regulares.

As asserções são mais bem utilizadas para validar parâmetros, propriedades, pré-condições, pós-condições, fluxo de controle da aplicação (para pontos do código que nunca deveriam ser alcançados) e, em geral, para verificação de erros que, de outra forma, você teria que comentar ou desabilitar quando o código estivesse pronto para produção.

Asserções também podem ser habilitadas para uma classe:

```
bash Copiar Editar

java -ea:NomeDaClasse ClassePrincipal
```

Ou para todas as classes em um pacote:

```
bash Copiar Editar

java -ea:com.exemplo ClassePrincipal
```

Para habilitar asserções no pacote não nomeado (quando você não usa instrução package):

```
bash Copiar Editar

java -ea:... ClassePrincipal
```

Ou, no caso raro de você querer habilitar asserções para as classes do sistema:

```
bash Copiar Editar

java -esa ClassePrincipal
```

Ou:

```
bash Copiar Editar

java -enablesystemassertions ClassePrincipal
```

Também há um comando para desabilitar as asserções:

```
bash Copiar Editar

java -da ClassePrincipal
```

Ou:

```
bash Copiar Editar

java -disableassertions ClassePrincipal
```

Com as mesmas opções que -ea. Para desabilitar para uma classe:

```
bash Copiar Editar

java -da:NomeDaClasse ClassePrincipal
```

Para desabilitar asserções em um pacote:

```
bash
java -da:com.exemplo ClassePrincipal
```

Para habilitar asserção no pacote não nomeado:

```
bash
java -ea:... ClassePrincipal
```

Para as classes do sistema:

```
bash
java -dsa ClassePrincipal
```

Ou:

```
bash
java -disablesystemassertions ClassePrincipal
```

Essa opção existe porque haverá momentos em que você desejará usar comandos como:

```
bash
java -ea -da:ClasseUm ClassePrincipal
```

Para habilitar asserções em todas as classes do programa, exceto para ClasseUm.

Pontos-chave

- Uma asserção é uma instrução usada para verificar se algo é verdadeiro e ajuda a detectar erros em um programa.
- Uma instrução assert tem a seguinte sintaxe:

```
java
assert expressãoBooleana;
assert expressãoBooleana: "Mensagem sobre o erro";
```

- Se a expressão booleana da instrução assert for avaliada como falsa, o programa lançará um `java.lang.AssertionError` e encerrará.
 - Por padrão, as asserções estão desabilitadas. Você pode usar os argumentos de linha de comando `-ea` (para habilitar) e `-da` (para desabilitar), com outras opções para classes, pacotes e classes do sistema ao executar o programa.
-

Autoavaliação

1. Dado:

```
java
public class Question_20_1 {
    public static void main(String[] args) {
        int x = 0;
        assert ++x > 0 : x;
    }
}
```

Qual é o resultado quando esse programa é executado com asserções habilitadas?

- A. 0
 - B. 1
 - C. Nada é impresso
 - D. Uma exceção é lançada em tempo de execução
-

2. Qual das seguintes afirmações é verdadeira?

- A. Você pode desabilitar asserções na linha de comando.
 - B. Asserções estão habilitadas por padrão nas classes do sistema.
 - C. Mesmo que as asserções estejam desabilitadas, se a expressão booleana for falsa, um erro de tempo de execução será lançado.
 - D. Dependendo da sintaxe, asserções podem lançar uma exceção verificada ou de tempo de execução.
-

3. Dado:

```
bash
java -esa -ea:com.exemplo ClassePrincipal
```

Qual das seguintes afirmações é verdadeira?

- A. Este comando habilita asserções em todas as classes do nosso programa.
 - B. Este comando habilita asserções nas classes do sistema em todas as classes do nosso programa.
 - C. Este comando habilita asserções nas classes do sistema apenas em ClassePrincipal.
 - D. Este comando habilita asserções nas classes do sistema dos pacotes com.exemplo.
-

4. Dado:

```
java
public class Question_20_4 {
    public static void main(String[] args) {
        assert isValid();
    }
    public static boolean isValid() {
        return false;
    }
}
```

Qual é o resultado quando esse programa é executado com asserções habilitadas?

- A. false
 - B. Nada é impresso
 - C. Erro de compilação
 - D. Um AssertionError é lançado em tempo de execução
-

5. Qual das seguintes instruções é equivalente a:

java

 Copiar  Editar

```
assert false : "Asserção é falsa";
```

- A. throw new Assertion("Asserção é falsa");
- B. throw new AssertionError("Asserção é falsa");
- C. if (false) throw new AssertionError("Asserção é falsa");
- D. if (false) throw new RuntimeException("Asserção é falsa");